Efficient Multiplier for pairings over Barreto-Naehrig Curves on Virtex-6 FPGA

Riadh Brinci*, Walid Khmiri[†], Mefteh Mbarek[‡], Abdellatif Ben Rabaâ, Ammar Bouallegue and Faouzi Chekir
National Engineering School of Tunis, ENIT
Tunis, Tunisia
* *br.riadh@gmail.com*
[†]*wkhmiri@yahoo.fr*
[‡]*mbarek19@gmail.com*

***Abstract*—This paper is devoted to the design of a 258-bit multiplier for computing pairings over Barreto-Naehrig (BN) curves at 128-bit security level. The proposed design is optimized for Xilinx field programmable gate array (FPGA). Each 258-bit integer is represented as a polynomial with five, 65 bit signed integer, coefficients . Exploiting this splitting we designed a pipelined 65-bit multiplier based on new Karatsuba-Ofman variant using non-standard splitting to fit to the Xilinx embedded digital signal processor (DSP) blocks. Our architecture is able to compute 258-bit multiplication suitable for BN curves using only 11 in-built DSP blocks available on Virtex-6 Xilinx FPGA devices. It is the least DSP blocks consumption in the known literature. This work can be extended to efficiently compute pairings at higher security levels.**

***Keywords*-Modular Multiplication, Modular Reduction, Cryptography, Pairing-Friendly Curves, Non-Standard Splitting, Field Programmable Gate Array(FPGA).**

I. Introduction

A bilinear pairing is a map $\mathbb{G}_1$ x $\mathbb{G}_2 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are typically additive groups and $\mathbb{G}_T$ is a multiplicative group and the map is linear in each component. Many pairings used in cryptography such as the Tate pairing [1], ate pairing [3], R-ate pairing [2] and optimal pairings [4], choose $\mathbb{G}_1$ and $\mathbb{G}_2$ to be specific cyclic subgroups of $E(\mathbb{F}_{p^k})$, and $\mathbb{G}_T$ to be a subgroup of $\mathbb{F}^*_{p^k}$.

A. Ate pairing

Let $\mathbb{F}_p$ be a finite field and let E be an elliptic curve defined over $\mathbb{F}_p$. Let r be a large prime dividing $\#E(\mathbb{F}_p)$ and k the embedding degree of $E(\mathbb{F}_p)$ with respect to r, namely, the smallest positive integer k such that $r|p^k-1$. For any finite extension field $\mathbb{K}$ of $\mathbb{F}_p$, denote with $E(\mathbb{K})[r]$ the $\mathbb{K}$-rational r-torsion group of the curve. For $P \in E(\mathbb{K})$ and an integer s,let O be the infinity point of E and $f_{s,P}$ be a $\mathbb{K}$-rational function or Miller function with divisor $(f_{s,P}) = s(P) - ([s]P) - (s-1)(O)$.

Let $\mathbb{G}_1 = E(\mathbb{F}_p)[r]$, $\mathbb{G}_2 = E(\mathbb{F}_{p^k}) \cap Ker(\pi_p - [p])$, where π_p is the p-th power Frobenius endomorphism;i.e $\pi_p : E \rightarrow E : (x, y) \mapsto (x^p, y^p)$ and $\mathbb{G}_T = \mu_r \subset \mathbb{F}^*_{p^k}$.

Let $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$ and let $t := p + 1 - \#E(\mathbb{F}_p)$ be the trace of Frobenius, then:

$$\alpha(P, Q) = (f_{t-1,Q}(P))^{(p^k-1)/r} \qquad (1)$$

is non-degenerate bilinear, and computable pairing, it is the ate pairing.

B. Pairing-Friendly Curves

An elliptic curve E over $\mathbb{F}_p$ is called pairing-friendly whenever there exists a large prime $r|\#E(\mathbb{F}_p)$ with $r > \sqrt{p}$ and the embedding degree k is small enough, e.g. $k < log_2(r)/8$. Many construction methods result in a parametrized family of elliptic curves, i.e. r and p are given by the evaluation of polynomials $r(u)$ and $f(u)$ at an integer value u. One of the most important examples of such families are the Barreto- Naehrig (BN) curves [5], ideally suited for implementing pairings at the 128-bit security level. These curves have $k = 12$ are defined by

$$p(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1 \qquad (2)$$

$$r(u) = 36u^4 + 36u^3 + 18u^2 + 6u + 1 \qquad (3)$$

for some $u \in \mathbb{Z}$ such that p is prime. We show that when choosing $u = 2^\tau + s$, where s is a reasonably small number, the modular multiplication in $\mathbb{F}_p$ can be substantially improved.

The R-ate pairing [2] is a generalization of the ate pairing and can be seen as an instantiation of optimal pairings [4]. Since the definition of the optimal ate pairing really depends on the particular elliptic curve one is using, we only provide the definition in the case of BN curves: using the same $\mathbb{G}_1$ and $\mathbb{G}_2$ as for the ate pairing, the optimal ate pairing on BN curves is defined as [7]

$$\varrho(P, Q) = (f.(f.l_{aQ,Q}(P))^p.l_{\pi(aQ+Q),aQ}(P))^{(p^k-1)/r} \qquad (4)$$

where $a = 6u + 2$, $f = f_{a,Q(P)}$ and $l_{A,B}$ denotes the line through points A and B.

C. FPGA resources

FPGA manufacturers integrate more and more of dedicated function blocks into modern devices. For example, Xilinx Virtex-6 FPGAs including separate columns of additional function hard cores for memory (BRAM) and arithmetic DSP operations. The DSP blocks are grouped in pairs that span the height of four or five CLBs, respectively.

Algorithm 1 Optimal Ate Pairing over BN curves [7]

```
Input: $a = |6u+2| = \sum_{i=0}^{s-1} a_i 2^i, P \in E(\mathbb{F}_p)[r], Q \in E(\mathbb{F}_{p^{12}})[r] \cap ker(\pi_p - [p])$
Output: $\varrho(Q,P) \in \mathbb{F}_{p^{12}}$
1: $T =\leftarrow Q$, $f \leftarrow 1$
2: for i=s-2 downto 0 do
3:     $T \leftarrow 2T$, $f \leftarrow f^2.l_{T,T}(P)$
4:     if $a_i = 1$ then
5:         $T \leftarrow T + Q$, $f \leftarrow f.l_{T,Q}(P)$
6:     end if
7: end for
8: $f \leftarrow (f.(f.l_{aQ,Q}(P))^p.l_{\pi(aQ+Q),aQ}(P))^{(p^k-1)/r}$
9: return $f$
```

The dual-ported BRAM matches the height of the pair of DSP blocks and supports a fast data path between memory and the DSP elements. Of particular interest is the use of these memory elements and DSP blocks for efficient boolean and integer arithmetic operations with low signal propagation time. Large devices of Xilinx Virtex-6 class are equipped with up to thousand individual function blocks of these dedicated memory and arithmetic units. Originally, the integrated DSP blocks as indicated by their name were designed to accelerate DSP applications, e.g., Finite Impulse Response (FIR) filters, etc. However, these arithmetic units can be programmed to perform universal arithmetic functions not limited to the scope of DSP filter applications; they support generic multiplication, addition and subtraction of (un)signed integers [14].

D. Outline

The remainder of this paper is organized as follows: Section II studies the most important existing works related to efficient hardware implementations of multiplication over $\mathbb{F}_p$ suitable for computing pairings over BN curves. Section III introduces our hardware design of 65 x 65 bit multiplier based on DSP macro for Virtex-6 and performance comparison. Section IV focuses on the hardware implementation and performance comparison of our 258 bit multiplier. Finally, section V provides conclusion and future works.

II. Related works

Since 2009 many hardware implementations of multiplication over $\mathbb{F}_p$ suitable for computing pairings on BN curves was described. The first work was described by Fan *et al.* [6]. Their proposed architecture was base on Hybrid Montgomery Multiplier (HMM) where multiplication and reduction was interleaved. In same year, a new Application Specific Integrated Circuit (ASIC) implementation of pairings over BN curves was proposed. In 2010 Fan *et al.* [7] proposed a new pipelined and parallelized version of their HMM [6]. In 2011, Corona *et al.* [8] proposed a new hardware implementation of 256bit multiplier suitable for computing pairings over BN curves. They used an asymmetric divide and conquer approach to efficiently implement their 65x65 bit multiplier. Their design used only 12 DSP slices on a Xilinx Virtex 6. In same year, Yao *et al.* [9] proposed a new hardware implementation of optimal ate pairing on Virtex 6. Their design computed multiplication over $\mathbb{F}_p$ using 32 DSP slices. They combined Lazy reduction with RNS representation.

III. Hardware design of 65x65 bit Multiplier

A. Karatsuba-Ofman Algorithm: Two-part splitting

Let X and Y be two integers to multiply, $X = \sum_{i=0}^{2n-1} x_i 2^i =$ and $Y = \sum_{i=0}^{2n-1} y_i 2^i$. We can split Y and Y into two parts:
$X = 2^n X_1 + X_0$ and $Y = 2^n Y_1 + Y_0$ The product $X.Y$ can be written as

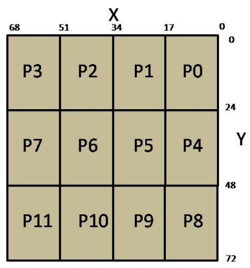
$$X.Y = 2^{2n} X_1 Y_1 + 2^n (X_1 Y1 + X_0 Y_0 - D_x D_y) + X_0 Y_0 \quad (5)$$

with $D_x = X_1 - X_0$ and $D_y = Y_1 - Y_0$. The same work can be expanded to three, four, five, six and seven-part splitting [10]. But these variants of Karatsuba-Ofman can not exploit the full performance of DSP blocks. The solution is to find a tilling method to fit to the Xilinx embedded DSP blocks.

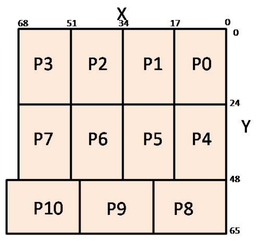
B. Asymmetric splitting

This section reported a work introduced for the first time in [11] and exploited in [8] to achieve 64bit multiplier for computing pairings over (BN) curves. These works used of the Virtex-6 25x18 signed multipliers. In this case, Y has to be decomposed into 24-bit chunks, while X is decomposed into 17-bit parts. This asymmetric variant of Karatsuba-Ofman algorithm consumed 12 DSP slices. This splitting would be optimal for a 72x68 product, but quite wasteful for the 65x65bit multiplication required for pairing computation. To get optimal exploitation of the in-built Virtex-6 resources [12] proposed a new splitting method called "non-standard tilling". This method is illustrated by figure 1 (b) and the following equation.

$$\begin{array}{rcc|c}
XY & = & X_{0:16}Y_{0:23} & P0 \\
 & + & 2^{17}(X_{17:33}Y_{0:23} & P1 \\
 & + & 2^{17}(X_{34:50}Y_{0:23} & P2 \\
 & + & 2^{17}X_{51:67}Y_{0:23})) & P3 \\
 & + & 2^{24}(X_{0:16}Y_{24:47} & P4 \\
 & + & 2^{17}(X_{17:33}Y_{24:47} & P5 \\
 & + & 2^{17}(X_{34:50}Y_{24:47} & P6 \\
 & + & 2^{17}X_{51:67}Y_{24:47}))) & P7 \\
 & + & 2^{48}(X_{0:23}Y_{48:64} & P8 \\
 & + & 2^{24}X_{24:47}Y_{48:64} & P9 \\
 & + & 2^{48}X_{48:71}Y_{48:64}) & P10
\end{array}$$



(a) Asymmetric tilling [11]



(b) Non standard tilling [12]

Figure 1. Tilling for 65x65 bit multiplication

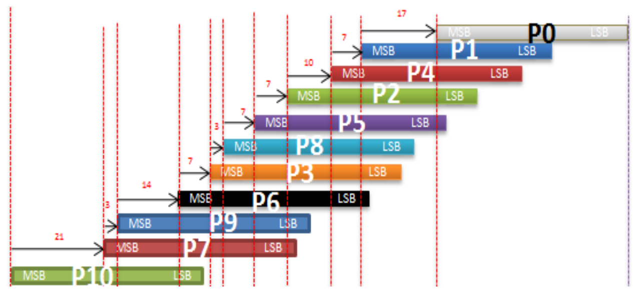


Figure 2. Partial Products regroupement

Table I
PERFORMANCE OF 65 BIT MULTIPLIER

	frequency	latency	DSP48 slices
[8]	300 MHz	4	12
this work	405 MHz	13	11

This multiplier was implemented, synthesized and test in VHDL for Xilinx Virtex-6 FPGA target. We have regrouped the Product P0. . . P10 as shown in figure 2. This arrangement let us exploit the shift, the multiplier and the adder offered by the in-built DSP blocks. Table 1 shows the performance of our design for 65 bit multiplier.

IV. ARCHITECTURE OF 258 BIT MULTIPLIER

In this section we present a hardware architecture for a modified version of HMM multiplier (algorithm 6) [7]. We choose $u = 2^{63} + 2^9 + 2^8 + 2^6 + 2^4 + 2^3 + 1$, this parameter generates BN curves suitable for pairings at 128 bit security level.

A. Modified HMM multiplier

Taking advantage of small number of sub-products done by equation 7 in [10] to compute product of quartic polynomials we propose the following algorithm 2. As mentioned in [7] we replaced multiplications by constants by shifts (power of 2) and additions. taking advantage of this the complexity of our modified HMM was reduced. Algorithm 2 is divided into 4 phases. Phase 1 computed the product $a.b$ using 13 sub-products (65x65 bit multipliers, 7x65 bit

Algorithm 2 Modified Parallel HMM

```
Input: $a = \sum_{i=0}^{4} a_i u^i$, $b = \sum_{i=0}^{4} b_i u^i$, $p = 36u^4 + 36u^3 + 24u^2 + 6u + 1$
Output: $r(u) = a.b.u^{-5} \ mod \ p$
    Phase1: Product of Quartic polynomials [10]
2:  Phase2: Coefficient Reduction
    for i=0 to 4 do
4:      $c_{i+1} \leftarrow c_{i+1} + (c_i \ div \ u)$, $c_i \leftarrow c_i \ mod \ u$
    end for
6:  Phase3: Polynomial Reduction
    $q(u) \leftarrow (-c_4 + 6(c_3 - 2c_2 - 6(c_1 - 9c_0)))u^4$
        $+(c_3 + 6(c_2 - 2c_1 - 6c_0))u^3$
        $+(-c_2 + 6(c_1 - 2c_0))u^2$
        $+(-c_1 + 6c_0)z.$
8:  $h(u) \leftarrow (36q_4)u^3$
        $+36(q_4 + q_3)u^2$
        $+12(2q_4 + 3(q_3 + q_2))u$
        $+6(q_4 + 4q_3 + 6(q_2 + q_1))u$
    $v(u) \leftarrow c(u)/u^5 + h(u)$
10: Phase4: Coefficient Reduction
    for i=0 to 3 do
12:     $v_{i+1} \leftarrow v_{i+1} + (v_i \ div \ u)$, $v_i \leftarrow v_i \ mod \ u$
    end for
14: return $r(u) \leftarrow v(u)$
```

multipliers and one 7x7 bit multiplier) and 22 addition [10]. Phase 2 and phase 4 are dedicated to the coefficient reduction using an expensive operation $div \ u$. But using the propriety of u written as $2^{63} + 857$ algorithm 5 in [7] replaced division by u by shifts additions and subtractions. The coefficient reduction is evaluated in parallel with sub-products. Phase 3 is the polynomial reduction it is also shifts and addition/subtraction operations. Our approach keep the circuit occupied all time.

B. implementation result of modified HMM

We used Xilinx Virtex-6 FPGA to implement our design it uses 11 DSP48 and 4 BRAMs to storage the coefficients of inputs output and partial variables. Our design takes 11

Table II
COMPARISON WITH EXISTING HARDWARE IMPLEMENTATIONS

criterion	[6]	[7]	[8]	[9]	[13]	this work
Frequency (MHz)	204	210	223	250	147	208
cycle per product	23	5	15	15	-	11
Latency	0	25	40	8	-	20
Device	ASIC	Virtex-6	Virtex-6	Virtex-6	Xilinx XCHVHX250T	Virtex-6
DSP48 Slices	-	46/288	12/48	36/2014	11/48	

cycles to achieve 258x258 multiplication with maximum frequency equal to 208 MHz. Table II compares our implementation with the state of art implementations. Our performance are competitive with others similar works.

V. CONCLUSION

This paper presented an original parallel and pipelined hardware implementation of modular multiplication. This implementation exploits the in-built Xilinx FPGA ressources to achieve a 258x258 bit multiplication using only 11 DSP blocks. This design is flexible and reconfigurable and can be used to efficiency compute pairings on BN curves at 128 bit security level or higher. As future work we will use this multiplier to implement optimal ate pairing on BN curves at 128 bit security level.

REFERENCES

[1] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott. Efficient Algorithms for PairingBased Cryptosystems. *In CRYPTO 2002, volume 2442 of Lecture Notes in Computer Science*, pages 354368. Springer, 2002.

[2] E. Lee, H. S. Lee, and C. M. Park. Efficient and Generalized Pairing Computation on Abelian Varieties. *In Cryptology ePrint Archive*, Available from http://eprint.iacr.org/. Report 2009/040.

[3] F. Hess, N. P. Smart, and F. Vercauteren. The Eta Pairing Revisited. *In IEEE Transactions on Information Theory, 52(10)* :pages 459-4602, Oct.2006.

[4] F. Hess. Pairing Lattices. *In Pairing 2008, volume 5209 of Lecture Notes in Computer Science*, pages 1838. Springer, 2008.

[5] P. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. *In Selected Areas in Cryptography, SAC 2005, LNCS 3897*, 319-331, 2006.

[6] J. Fan, F. Vercauteren, and I. Verbauwhede. Faster Fp Arithmetic for Cryptographic Pairings on Barreto-Naehrig Curves. *In CHES 2009, volume 5747 of Lecture Notes in Computer Science*, pages 240-253. Springer, 2009.

[7] J. Fan, F. Vercauteren, and I. Verbauwhede. Efficient hardware implementation of Fp-arithmetic for pairing-friendly curves. *In IEEE Transaction ON Computers*, PP(99):1 2011.

[8] C. C. Corona, E. F. Moreno and F. R. Henriquez. Efficient hardware implementation of Fp-arithmetic for pairing-friendly curves. *In IEEE Transaction ON Computers*, PP(99):1 2011.

[9] G. X. Yao, J. Fan, R. C. Cheung, and I. Verbauwhede. A high speed pairing coprocessor using RNS and lazy reduction. *Cryptology ePrint Archive*, Available from http://eprint.iacr.org/. Report 2011/258, 2011.

[10] P. L. Montgomery. Five, six, and seven-term Karatsuba-like formulae, *In IEEE Transactions on Computers*, vol. 54, no. 3, pp. 362369, 2005.

[11] S. Srinath and K. Compton. Automatic generation of high performance multipliers for FPGAs with asymmetric multiplier blocks. *In Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays, FPGA 10*, pages 5158, New York, NY, USA, 2010. ACM.

[12] F . de Dinechin and B. Pasca. Large multipliers with fewer DSP blocks , *In Field Programmable Logic and Applications*. IEEE,Aug. 2009

[13] A. Mondal, S. Ghosh, A. Das, D. R. Chowdhury. Efficient FPGA Implementation of Montgomery Multiplier Using DSP Blocks. In *Progress in VLSI Design and Test - 16th International Symposium, VDAT 2012*, Lecture Notes in Computer Science 7373 Springer 2012: 370-372.

[14] T. Geuneysu. Utilizing hard cores of modern FPGA devices for high-performance cryptography. In *Journal of Cryptographic Engineering*, 1:3755, 2011